

Les listes avec le langage PYTHON – Partie 1

1. Déclaration d'une liste (tableau)

```
Numbers = [1,2,3,4]
```

2. Déclaration d'une liste vide

```
maListe = []
```

3. Ajouter un élément à ma liste

```
maListe.append(valeur)
```

Remarque : la liste peut contenir des valeurs de types différents

```
Liste = [ 1, "bonjour", true, 20.]
```

4. Insérer un élément à la liste

```
maListe.insert(indice,valeur)
```

5. Supprimer un élément de la liste

```
del Numbers[1] # je supprime la valeur d'indice 1
```

```
Numbers.remove(2) # je supprime la valeur 2
```

6. La taille de la liste

```
Print(len(Numbers))
```

7. Afficher une liste

Affichage par défaut

```
Print(Numbers)
```

boucle sur les valeurs

```
For n in Numbers :
```

```
    Print(n)
```

Boucle sur les indices

```
for i in range( len(Numbers)) :
```

```
    print(Numbers[i])
```

8. Les indices négatifs

Afficher la dernière valeur

```
Print ( Numbers [ len(Numbers) – 1 ] )
```

Python simplifie cette écriture

```
Print ( Numbers [ -1 ] )
```

Afficher la première valeur

```
Print( Numbers[ 0 ] ) = Print ( Numbers [ len(Numbers) – len(Numbers) ] )  
= Print ( Numbers [ - len(Numbers) ] )
```

Exercice :

Demander à l'utilisateur de saisir une liste de 10 entiers, et inverser les valeurs du tableau en utilisant la permutation

23	56	33	8	1	23	66	7	44	79
0	1	2	3	4	5	6	7	8	9

Permuter(t[0] avec t[9])

79	56	33	8	1	23	66	7	44	23
-----------	----	----	---	---	----	----	---	----	-----------

Permuter(t[1] avec t[8])

79	44	33	8	1	23	66	7	56	23
-----------	-----------	----	---	---	----	----	---	-----------	-----------

Permuter(t[2] avec t[7])

79	44	7	8	1	23	66	33	56	23
-----------	-----------	----------	---	---	----	----	-----------	-----------	-----------

Permuter(t[3] avec t[6])

79	44	7	66	1	23	8	33	56	23
-----------	-----------	----------	-----------	---	----	----------	-----------	-----------	-----------

Permuter(t[4] avec t[5])

79	44	7	66	23	1	8	33	56	23
-----------	-----------	----------	-----------	-----------	----------	----------	-----------	-----------	-----------

T[0],T[9] = T[9], T[0]

T[1],T[8] = T[8], T[1]

T[2],T[7] = T[7], T[2]

T[3],T[6] = T[6], T[3]

T[4],T[5] = T[5], T[4]

10 entier : 5 permutations

5 entier : 2 permutations

Je vais faire une boucle Pour (10/2) fois

Solution

```
# Les entrées
liste = []
for i in range(10):
    #print ("donner la valeur",i+1)
    liste.append(int(input(f"donner la valeur {i+1}")))

#le traitement
taille = len(liste) #taille = 10
for i in range(taille//2):
    liste[i],liste[ -1 - i] = liste[ -1 - i] , liste[i]

# sortie
print(liste)
```

9. Inverser une liste

```
liste.reverse()

print(liste)
```

10. Recopier une liste dans une autre liste :

A Retenir :

- Le nom d'une liste pointe sur son adresse
- Le nom d'une variable pointe sur sa valeur

```
T1 = [10,20,30]
T2 = T1    #T2 prend l'adresse de T1, T1 et T2 ont la
           #même zone mémoire
```

```
T2[0] = 100
print(T1[0]) #100
print(T2[0]) #100
```

```
A = 5
B = A    #B prend la valeur de A
        #A et B ont deux zones mémoires différentes
```

```
B=10
```

```
Print(A) #5
Print(B) #10
```

Nom Variable	Adresse Mémoire	Valeur
T1,T2	A0001	100
		20
		30
A	A0012	5
B	A0016	5- 10

Comment créer une nouvelle liste à partir d'une ancienne liste ?

Il faut utiliser les tranches.

```
T=[10,20,30,40,50]
#T1 = T[0:5] #(tout le tableau )
T1 = T[:] #(tout le tableau ) de l'indice 0 jusqu'à la fin
print(T)
```

```
print(T1)
T[0]=10
print(T)
print(T1)
```

Remarque : les valeurs de T n'ont pas changé

```
T2=T[2:] #les indices de 2 jusqu'au dernier: les indices 2,3,4
T3=T[:3] #les indices: 0,1,2
T4=T[-2:]#à partir de l'indice taille-2 (les 2 dernières valeurs)
T5 = T[3:5] # les indices 3,4

del T[2:4] # je supprime les indice 2 et 3
del T [:] # je vide tout le tableau
print(T)
del T # je supprime le tableau
print(T) #erreur, T n'existe plus
```

11. Les opérateurs in et not in

Syntaxe général :

(valeur in myList)

(valeur not in myList)

renvoie True or False

```
T=[1,2,3,4,5,6]
print( 6 in T) #True
print(10 in T) # False
print(11 not in T) #True
print(5 not in T) #False
```